I DON'T CARE

i am yangqi

contents

1 nlp-blog

关键词：

自然语言处理（NLP）.词向量（Word Vectors）.奇异值分解(Singular Value Decomposition). Skip-gram. 连续词袋（CBOW）,负采样样本（Negative Sampling)

1.1 词向量

What: 词组用向量表示
Why:
1.NLP转为ML问题，第一步就是将符号数学化
2词向量编码词组,使其代表N维空间中的一个点，点与点之间距离可以代表深层信息。每一个词向量的维度都可能会表征一些意义（物理含义）。例如，语义维度可以用来表明时态（过去与现在与未来），计数（单数与复数），和性别（男性与女性）
How:编码方式：比如one-hot vector

1.1.1 one-hot vector

What:对词库中n个词，每个词在某个index下取到1，其余位置为0
Disadvantge:
1.维数灾难
2.词向量无法表示词组相似性：$(W^{hotel})^T w^{motel} = (W^{hotel})^T w^{cat} = 0$[hotel和motel是近义词]。
Improve:可以把词向量的维度降低一些，在这样一个子空间中，可能原本没有关联的词就关联起来了

1.2 构造词向量方法-基于SVD

How：遍历所有的文本数据集，然后统计词出现的次数，接着用一个矩阵X来表示所有的次数情况，紧接着对X进行奇异值分解得到一个USV^T的分解。然后用U的行（rows）作为所有词表中词的词向量。对于矩阵X,有如下方法：

1.2.1 词-文档矩阵

What：行：文档M。列：词组V。
How：遍历文件，词组i出现在文件j中，将Xij值加一。得到矩阵R|V|×M

1.2.2 基于窗口的共现矩阵

What：同上，将词频换成了相关性矩阵
How: 固定大小窗口，统计每个词出现在窗口中次数。
例如： I enjoy flying. | | I like NLP. | | I like deep learning.

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 1: 基于窗口共现矩阵

1.2.3 奇异值分解

What:将矩阵用更小更简单的子矩阵的相乘来表示【机器学会抽取重要特征】
Why:降维
Example:PCA、数据(图像)压缩、搜索引擎语义层次检索LSI
How: $A = U\Sigma V^T$ 并根据保留百分比保留k个维度. 【奇异值σ跟特征值类似，在矩阵∑中也是从大到小排列，而且σ的减少特别的快，在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上了。也就是说，我们也可以用前r大的奇异值来近似描述矩阵】
Result: U作为词嵌入矩阵，对于词表中的每一个词，都用一个k维的向量表示

2 algorithm

2.1 算法

3 machine-learning

3.1 学习算法

3.1.1 监督学习

What：数据集中每个样本都有“正确答案”，再根据样本作出预测
Example:
回归问题：
What：推导出连续的输出。
Example：房价分析／销量预测
分类问题：
What：推导出离散的输出。
Example：乳腺肿瘤判断／垃圾邮件问题

3.1.2 无监督学习

What：交给算法大量数据，让算法为我们从数据中找出某种结构
Example:
聚类问题：
How：谷歌news。同一主题的聚类

3.2 第一个学习算法-单变量线性回归

What：只有一个特征(输入变量)
回归：
What:根据之前的数据预测一个准确输出值
线性回归：
What: 确定两种或两种以上变量间相互依赖的定量关系。$y = w'x + e$
How:

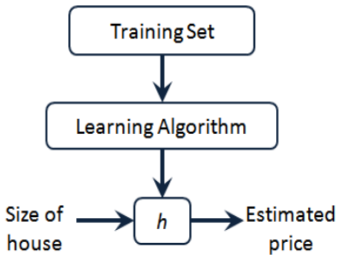

Figure 2: 监督学习算法工作方式

h的一种可能表达方式：$h_\theta(x) = \theta_0 + \theta_1 x$ 因为只有一个特征(输入变量)，这样的问题叫做单变量线性回归问题.

3.2.1 代价函数-用来求解线性回归方程参数

What:平方误差函数（平方误差代价函数）。
How:建模误差的平方和: $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^i)) - y^i)^2$
目标：$\text{Min}\, J(\theta_0, \theta_1)$
How：
1. 代价函数(等高线图):在三维空间中存在一个值使得$J(\theta_0, \theta_1)$最小
2. 需要算法【自动】找出使得J 最小化的θ_0, θ_1的值

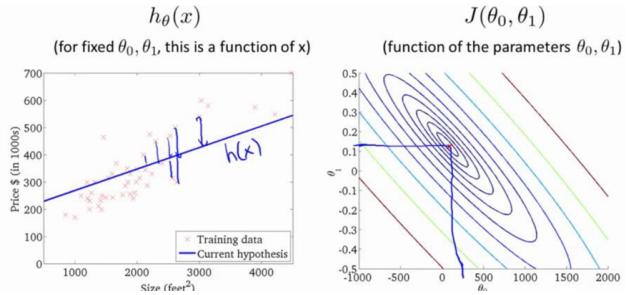

Figure 3: 代价函数-等高线图

3.2.2 梯度下降-求代价函数最小值

What:求函数最小值的算法
How：随机选择一个参数的组合($\theta_0, \theta_1, \theta_2, \theta_3$)，计算代价函数，然后寻找下一个能让代价函数值下降最多的参数组合。直到到达一个局部最小值。【由于没有常识所有的参数组合，不能保证局部最小值】